Facial Deformations for MPEG-4

Marc Escher, Igor Pandzic, Nadia Magnenat Thalmann

MIRALab - CUI
University of Geneva
24 rue du Général-Dufour
CH1211 Geneva 4, Switzerland
{Marc.Escher, Igor.Pandzic, Nadia.Thalmann}@cui.unige.ch
http://miralabwww.unige.ch/

Abstract

The new MPEG-4 standard, scheduled to become an International Standard in February 1999, will include support not only for natural video and audio, but also for synthetic graphics and sounds. In particular, representation of human faces and bodies will be supported. In the current draft specification of the standard [MPEG-N1901, MPEG-N1902] Facial Animation Parameters (FAPs) and Facial Definition Parameters (FDPs) are defined. FAPs are used to control facial animation at extremely low bitrates (approx. 2 kbit/sec). FDPs are used to define the shape of the face by deforming a generic facial model, or by supplying a substitute model. We present algorithms to interpret the part of FDPs dealing with the deformation of a generic facial model, leading to a personalisation of the model. The implementation starts from a generic model, which is deformed in order to fit the input parameters. The input parameters must include the facial feature points, and may optionally include texture coordinates, and a calibration face model. We apply a cylindrical projection to the generic face in order to interpolate any missing feature points and to fit the texture, if supplied. Then we use a Dirichlet Free Form Deformation [Moccozet 97] interpolation method to deform the generic head according to the set of feature points. If the calibration face model is present, the fitting method is based on cylindrical projections matching and barycentric coordinates to interpolate the non-feature points.

Keywords: MPEG-4, SNHC, Facial animation, Face modelling.

1. Introduction

ISO/IEC JTC1/SC29/WG11 (Moving Pictures Expert Group - MPEG) is currently working on the new MPEG-4 standard [Koenen97, MPEG-N1901, MPEG-N1902], scheduled to become International Standard in February 1999. In a world where audio-visual data is increasingly stored, transferred and manipulated digitally, MPEG-4 sets its objectives beyond "plain" compression. Instead of regarding video as a sequence of frames with fixed shape and size and with attached audio information, the video scene is regarded as a set of dynamic objects. Thus the background of the scene might be one object, a moving car another, the sound of the engine the third etc. The objects are spatially and temporally independent and therefore can be stored, transferred and manipulated independently. The composition of the final scene is done at the decoder, potentially allowing great manipulation freedom to the consumer of the data.

Video and audio acquired by recording from the real world is called natural. In addition to the natural objects, synthetic, computer generated graphics and sounds are being produced and used in ever increasing quantities. MPEG-4 aims to enable integration of synthetic objects within the scene. It will provide support for 3D Graphics, synthetic sound, Text to Speech, as well as synthetic faces and bodies. In this paper we concentrate on the representation of faces in MPEG-4, and in particular the methods to produce personalised faces from generic faces.

The following section provides the introduction to the representation of faces in MPEG-4. We explain how Facial Animation Parameters and Facial Definition Parameters are used to define the shape and animation of faces. In section 3 we present our algorithm for the interpretation of Facial Definition Parameters. In the final sections we present the results and conclusions, as well as the ideas for future work.

2. Faces in MPEG-4

The Face and Body animation Ad Hoc Group (FBA) deals with coding of human faces and bodies, i.e. efficient representation of their shape and movement. This is important for a number of applications ranging from communication, entertainment to ergonomics and medicine. Therefore there exists quite a strong interest for standardisation. The group has defined in detail the parameters for both definition and animation of human faces and bodies. This draft specification is based on proposals from several leading institutions in the field of virtual humans research. It is being updated within the current MPEG-4 Committee Draft [MPEG-N1901, MPEG-N1902].

Definition parameters allow detailed definition of body/face shape, size and texture. Animation parameters allow to define facial expressions and body postures. The parameters are designed to cover all naturally possible expressions and postures, as well as exaggerated expressions and motions to some extent (e.g. for cartoon characters). The animation parameters are precisely defined in order to allow accurate implementation on any facial/body model.

In the following subsections we present in more detail the Facial Animation Parameters (FAPs) and the Facial Definition Parameters (FDPs).

2.1 Facial Animation Parameter set

The FAPs are based on the study of minimal facial actions and are closely related to muscle actions. They represent a complete set of basic facial actions, and therefore allow the representation of most natural facial expressions. The lips are particularly well defined and it is possible to precisely define the inner and outer lip contour. Exaggerated values permit actions that are normally not possible for humans, but could be desirable for cartoon-like characters.

All the parameters involving translational movement are expressed in terms of the Facial Animation Parameter Units (FAPU). These units are defined in order to allow interpretation of the FAPs on any facial model in a consistent way, producing reasonable results in terms of expression and speech pronunciation. They correspond to fractions of distances between some key facial features (e.g. eye distance). The fractional units used are chosen to allow enough precision.

The parameter set contains two high level parameters. The viseme parameter allows to render visemes on the face without the need to express them in terms of other parameters or to enhance the result of other parameters, insuring the correct rendering of visemes. Similarly, the expression parameter allows definition of high level facial expressions.

2.2 Facial Definition Parameter set

An MPEG-4 decoder supporting the Facial Animation must have a generic facial model capable of interpreting FAPs. This insures that it can reproduce facial expressions and speech pronunciation. When it is desired to modify the shape and appearance of the face and make it look like a particular person/character, FDPs are necessary.

The FDPs are used to personalise the generic face model to a particular face. The FDPs are normally transmitted once per session, followed by a stream of compressed FAPs. However, if the decoder does not receive the FDPs, the use of FAPUs insures that it can still interpret the FAP stream. This insures minimal operation in broadcast or teleconferencing applications.

The Facial Definition Parameter set can contain the following:

- 3D Feature Points
- Texture Coordinates for Feature Points (optional)
- Face Scene Graph (optional)
- Face Animation Table (FAT) (optional)

The Feature Points are characteristic points on the face allowing to locate salient facial features. They are illustrated in

Figure **1**. Feature Points must always be supplied, while the rest of the parameters are optional.

The Texture Coordinates can be supplied for each Feature Point.

The Face Scene Graph is a 3D-polygon model of a face including potentially multiple surfaces and textures, as well as material properties.

The Face Animation Table (FAT) contains information that defines how the face will be animated by specifying the movement of vertices in the Face Scene Graph with respect to each FAP as a piecewise linear function. We do not deal with FAT in this paper.

The Feature Points, Texture Coordinates and Face Scene Graph can be used in four ways:

- If only Feature Points are supplied, they are used on their own to deform the generic face model.
- If Texture Coordinates are supplied, they are used to map the texture image from the Face Scene Graph on the face deformed by Feature Points. Obviously, in this case the Face Scene Graph must contain exactly one texture image and this is the only information used from the Face Scene Graph.
- If Feature Points and Face Scene Graph are supplied, and the Face Scene Graph contains a non-textured face, the facial model in the Face Scene Graph is used as a Calibration Model. All vertices of the generic model must be aligned to the surface(s) of the Calibration Model.
- If Feature Points and Face Scene Graph are supplied, and the Face Scene Graph contains a textured face, the facial model in the Face Scene Graph is used as a Calibration Model. All vertices of the generic model must be

aligned to the surface(s) of the Calibration Model. In addition, the texture from the Calibration Model is mapped on the deformed generic model.

In the following section we describe how these options are supported in our system

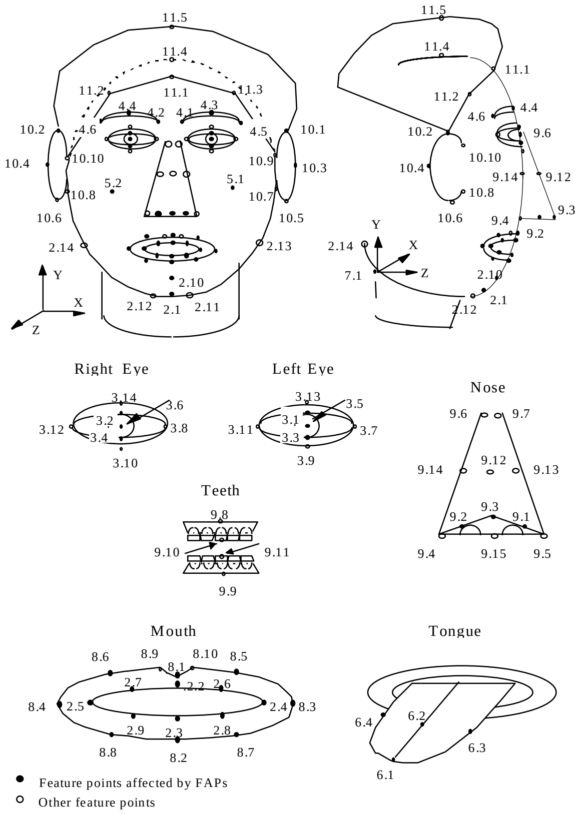


Figure 1: FDP feature point set

3. Algorithms for interpretation of FDPs

3.1 Interpretation of Feature Points only

The first step before computing any deformation is to define a generic head that can be deformed efficiently to any humanoid head by moving specific feature points. The model we use is a 3D polygonal mesh composed of approx. 1500 vertices on which we have fixed a set of vertices that correspond to the feature points defined in MPEG. (Figure2).

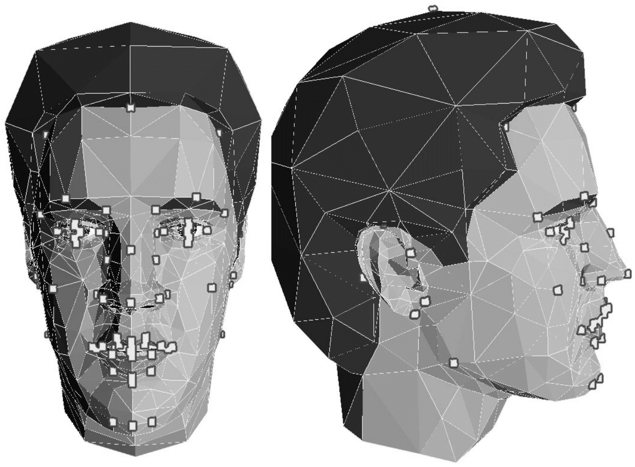


Figure 2: Generic model and feature points

The deformation (fitting) of the generic head is computed using a Dirichlet Free Form Deformation method, which allow a volume deformation using control points while keeping the surface continuity. This method has been developed in MIRALab [Moccozet 97] and uses a Dirichlet diagram to compute the Sibson's local coordinates for the non-feature points interpolation. Figure 3 shows a deformation of the chin on the generic model by dragging the four control points of the chin (dark boxes). Light boxes represent control points.

3.1.1 Missing feature point interpolation

As the Sibson's coordinates calculation is a heavy computation process, it is performed only once for each generic head and saved as a data file. This restrains the use of the DFFD method only to the case when all feature points are available, which may not always be the case. Therefore we perform a pre-processing to interpolate the missing feature points. A cylindrical projection of all the feature points of the generic face, and a Delaunay triangulation of the encoded points are computed. Barycentric coordinates are then calculated for the non-given feature points. Each feature point that had no 3D FDP coordinate at the encoder has now 3 values corresponding each one to the weight of a bounding feature point vertex.

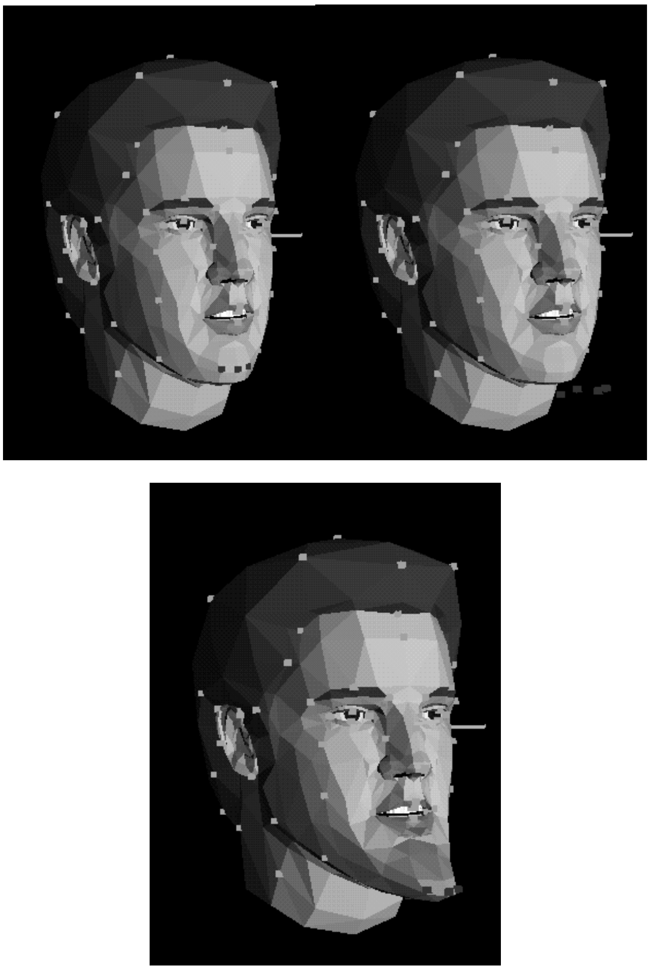


Figure 3: DFFD example

The FDP interpolated coordinate is:

Xf = Xi+Wa *(Xfa-Xia)+Wb * (Xfb-Xib)+Wc * (Xfc-Xic)

Where:

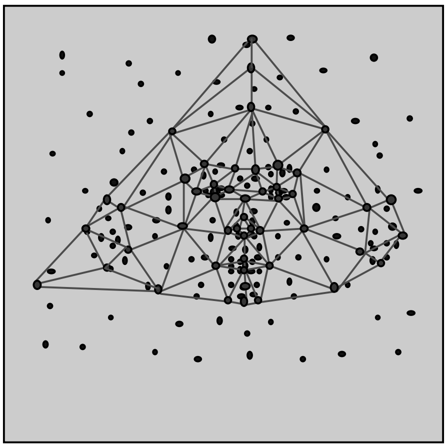
Xf = final 3D coordinate of the non encoded feature point
Xi = initial 3D coordinate of the non encoded feature point
Wa,b,c = barycentric coordinate
Xfa,b,c = final 3D coordinate of the 3 bounding vertices
Xia,b,c = initial 3D coordinate of the 3 bounding vertices

Once the 3D position of all the feature points are known we apply the DFFD method to fit the generic head to the extracted/interpolated FDP points.

3.2 Interpretation of Feature Points and Texture

The method we use for computing the texture coordinates uses a cylindrical projection of all the points of the generic 3D face instead of a planar projection. The use of cylindrical projection allows all the points of the head to be texture mapped. Even if generally only one front picture is given as a texture image and only the front part of the face is textured, it is always better to have a more general method that allows a complete mapping of the head.

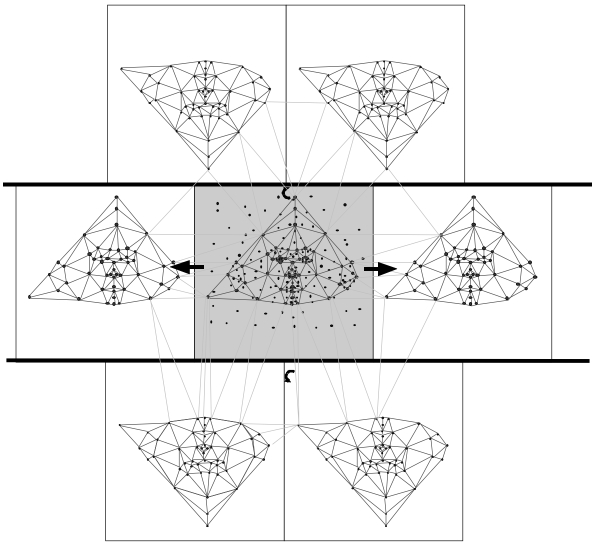
The problem with the cylindrical projection is that the Delaunay triangulation of the projected feature points doesn’t include all the non-feature points. (Figure 4.)



Linked dots: Projected feature points
Unlinked dots: Projected non-feature points
Green lines: Feature points triangulation

Figure 4. Cylindrical projection of the head points

This problem can be resolved if we use the property of continuity of the cylindrical projection and some neighbourhood approximation to generate a convex Delaunay triangulation. We use these properties to develop an method that include all the non-feature points in a triangulation of feature points (Figure 5)



Linked dots: Projected feature points

Unlinked dots:	Projected non-feature points
Dark lines:	Feature points triangulation
Light lines:	Expanded triangulation

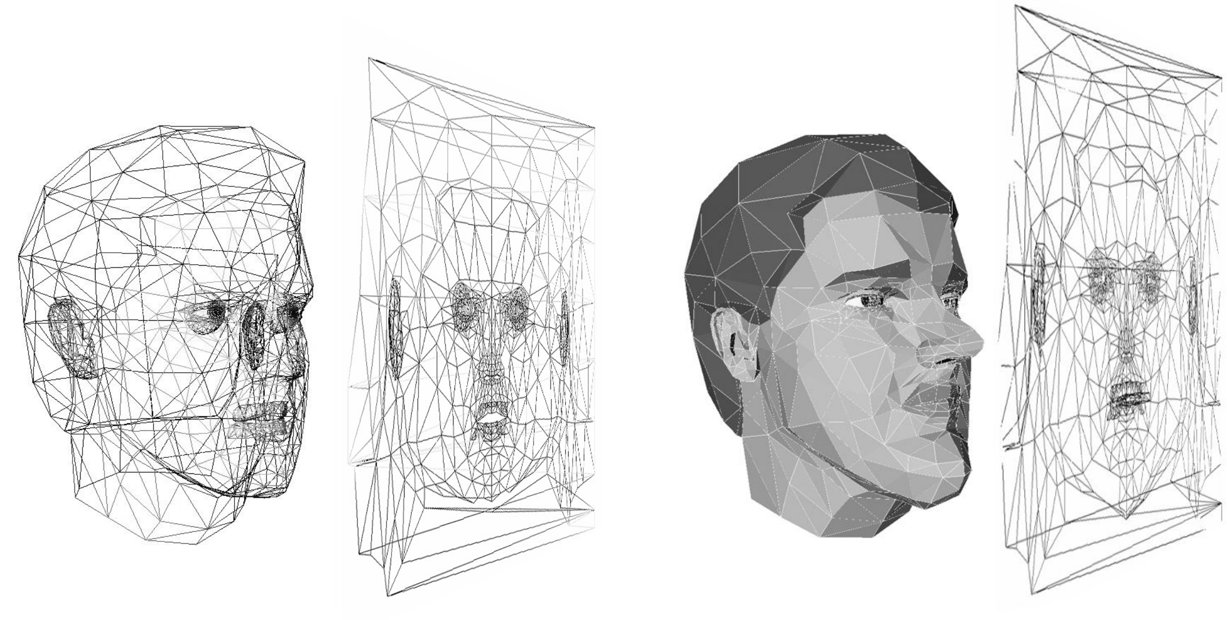
Figure 5: Expansion of the feature points

Basically the feature points are duplicated on the left and right side by a horizontal shift. The upper and lower parts are filled with 2 duplications each, using a horizontal symmetry. An “expanded” Delaunay triangulation is then performed, it now includes all the non-feature points. This method which approximate a spherical projection gives visually acceptable results. (Figure 5)

3.3 Interpretation of Feature Points and Calibration Model

In this profile, a 3D-calibration mesh is given along with the position of its control points. The goal is to fit the generic mesh on the calibration one. Our method starts with the cylindrical projection of both 3D meshes (Figure 6).

The next step is to map the projection of the generic map on the projection of the calibration one. The procedure is exactly the same as the one previously described for the texture fitting, with the use of the 3D projected feature points except of the 2D texture feature points. When the 2D projection of the generic mesh is fitted on the calibration one, we compute the barycentric coordinates of every non-feature points of the generic head in relation with the triangulation of the calibration mesh. At this stage every point of the generic mesh is either a feature point with a corresponding new 3D location, or a non-feature point with barycentric coordinates. The new 3D position of the non-feature points is interpolated using the formula expressed in 3.1. This method works fine for most of the face surface, but for specific regions with high complexity such as the ears, some distortions may appear.



Generic head cylindrical projection

Calibration model cylindrical projection

Figure 6: Cylindrical projection

3.4 Interpretation of Feature Points and texture and Calibration Model

The addition of texture is done in the same way as described in 3.1. I.e. Expantion and triangulation of the texture feature points, local barycentric coordinates extraction for every non-feature points of the 3D-head mesh. Concatenation of a mouth and eye picture on the texture image in order to apply texture on hidden parts. (Figure 7) The iris colour of the eyes is selected automatically from the texture picture by extraction the HLS parameters from the most open eye. The concatenated eye picture is then modified to match these parameters.

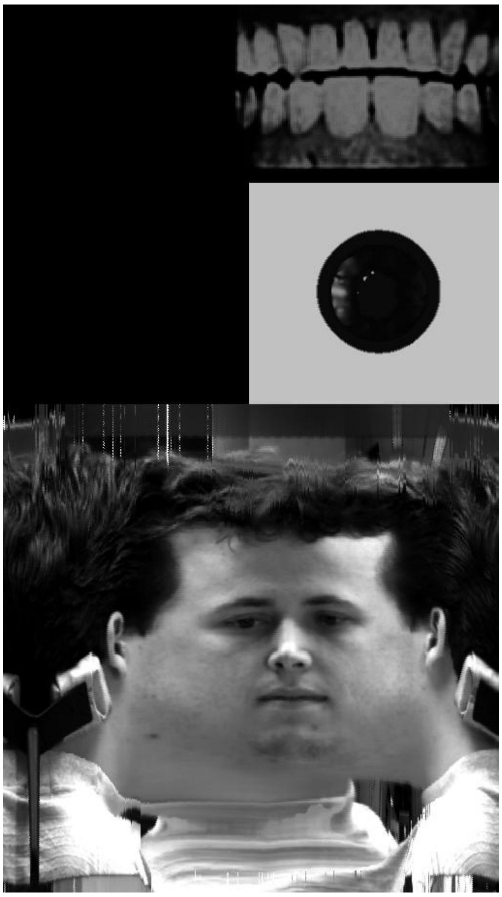


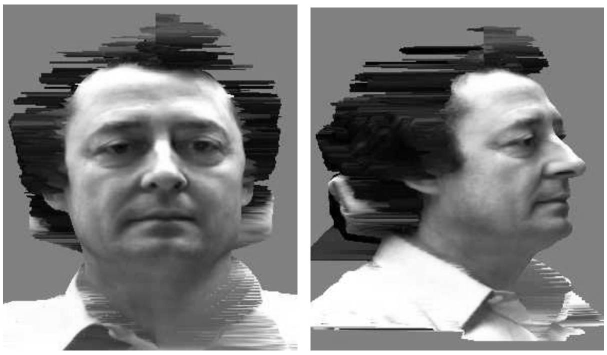
Figure 7: Complete texture image

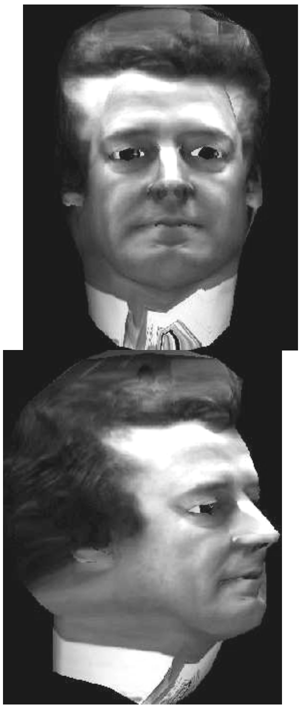
4. Results

We were involved in the CE FBA3. The experiment was carried out with test FDP and texture files all available from the FBA Core Experiments home page:

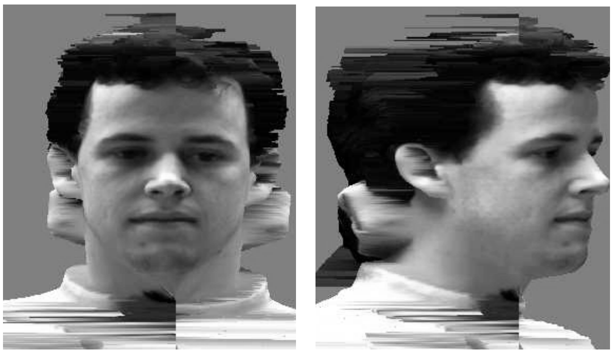
- Jim_n.fdp, Jim_n.fdp.text, Jim_n.color
- Claude_n.fdp,Claude_n.fdp.text, Claude_n.color
- Chen_n.fdp, Chen_n.fdp.text, Chen_n.color

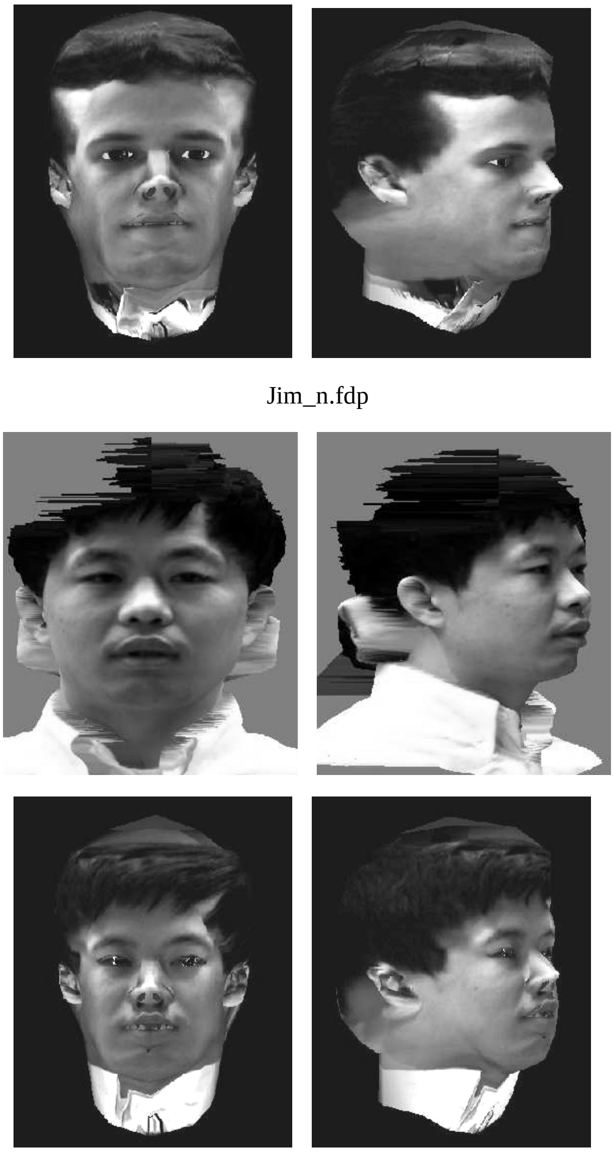
The results are shown in the following pictures:





Claude_n.fdp





Jim_n.fdp

Chen_n.fdp

Figure 8: Final results

5. Conclusions

This papers has described some techniques of face fitting and texturing adapted to the actual definitions of the MPEG-4 SNHC Face Definition Parameters. We have presented our implementation of texturing using cylindrical projection and in particular a method for generating an encompassing delaunay triangulation by expanding the projected feature points. The face modelling using 3D feature points or a calibration model, using extensively delaunay triangulation and barycentric coordinates has also been explained. Finally some result have been shown.

6. Acknowledgements

This research is financed by the ACTS project AC057 VIDAS.

7. References

[Boulic 95] Boulic R., Capin T., Huang Z., Kalra P., Lintermann B., Magnenat-Thalmann N., Moccozet L., Molet T., Pandzic I., Saar K., Schmitt A., Shen J., Thalmann D., "The Humanoid Environment for Interactive Animation of Multiple Deformable Human Characters", *Proceedings of Eurographics '95*, 1995.

[Kalra 92] Kalra P., Mangili A., Magnenat Thalmann N., Thalmann D., "Simulation of Facial Muscle Actions Based on Rational Free Form Deformations", *Proc. Eurographics '92*, pp.59-69., 1992.

[Kalra 93] Kalra P. "An Interactive Multimodal Facial Animation System", *PhD Thesis nr. 1183*, EPFL, 1993.

[Koenen 97] Koenen R., Pereira F., Chiariglione L., "MPEG-4: Context and Objectives", *Image Communication Journal, Special Issue on MPEG-4*, Vol. 9, No. 4, May 1997.

[Moccozet 97] Moccozet L. Magnenat Thalmann N., “Dirichlet Free-Form Deformation and their Application to Hand Simulation”, *Proc. Computer Animation ’97, IEEE Computer Society*, pp.93-102.

[MPEG-N1901] "Text for CD 14496-1 Systems", ISO/IEC JTC1/SC29/WG11 N1886, MPEG97/November 1997.

[MPEG- N1902] "Text for CD 14496-2 Video", ISO/IEC JTC1/SC29/WG11 N1886,